Complex & Intelligent Systems (2023) 9:3989–4002
https://doi.org/10.1007/s40747-022-00944-x

ORIGINAL ARTICLE

Attention-guided video super-resolution with recurrent multi-scale spatial–temporal transformer

Wei Sun[1,2] · Xianguang Kong[3] · Yanning Zhang[4]

Received: 24 May 2022 / Accepted: 26 November 2022 / Published online: 15 December 2022

Abstract
Video super-resolution (VSR) aims to recover the high-resolution (HR) contents from the low-resolution (LR) observations relying on compositing the spatial–temporal information in the LR frames. It is crucial to propagate and aggregate spatial–temporal information. Recently, while transformers show impressive performance on high-level vision tasks, few attempts have been made on image restoration, especially on VSR. In addition, previous transformers simultaneously process spatial–temporal information, easily synthesizing confused textures and high computational cost limit its development. Towards this end, we construct a novel bidirectional recurrent VSR architecture. Our model disentangles the task of learning spatial–temporal information into two easier sub-tasks, each sub-task focuses on propagating and aggregating specific information with a multi-scale transformer-based design, which alleviates the difficulty of learning. Additionally, an attention-guided motion compensation module is applied to get rid of the influence of misalignment between frames. Experiments on three widely used benchmark datasets show that, relying on superior feature correlation learning, the proposed network can outperform previous state-of-the-art methods, especially for recovering the fine details.

Keywords Video super-resolution · Spatial-temporal transformer · Attention mechanism · Motion compensation

Introduction

Since the limited resolution of long-distance imaging and imaging devices in specific scenes, generating high-resolution (HR) images from original low-resolution (LR) content is under great demand. The aim of video super-resolution (VSR) is to gather complementary information across LR video frames for reconstructing the HR details. As a fundamental task in computer vision, VSR is usually adopted to enhance visual quality, which has great value in our daily life, such as security and surveillance imaging system [1], high-definition television [2], etc.

Video super-resolution (VSR) is challenging partly because recovering details from LR observations is highly ill-posed as many inverse problems. Previous prior-based methods most focus on pixel-level motion and blur kernel estimation, trying to solve a complex optimization problem [3]. With the recent development on deep learning, convolutional neural networks (CNNs)-based VSR learn the complicated up-sampling relations between LR and HR videos on huge datasets. Compared to the traditional approaches, these significantly improve performance.

From a methodology perspective, unlike single image super-resolution (SISR) [4,5] that usually concentrated on learning on spatial dimensions, VSR mainly concerns temporal information. Although a straightforward strategy for VSR is to run SISR frame by frame, SISR ignores the temporal correlations, which leads to discontinuity in reconstruction, resulting in the flickering artifact. Therefore, a key issue to the success of VSR is how to aggregate information from multiple highly related but misaligned frames [6].

✉ Yanning Zhang
ynzhang@nwpu.edu.cn

Wei Sun
sunwei052@gmail.com

Xianguang Kong
kongxianguang@xidian.edu.cn

[1] School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an, China

[2] Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing, Xi'an, China

[3] School of Mechano-Electronic Engineering, Xidian University, Xi'an, China

[4] School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, China

One prevalent solution is the space-based framework [7,8], where each LR frame in the video is restored using the fusion information within adjacent frames. In contrast to the space-based framework, a time-based framework [9,10] attempts to exploit long-temporal dependencies within time-series data based upon recurrent mechanism. In general, compared to space-based frameworks, time-based frameworks allow a more compact model. Nevertheless, the problem of communicating long-term information under misaligned situations remains formidable.

To address the information transfer issue, Fuoli et al. [11] proposes a recurrent latent space propagation algorithm, high dimensional latent states are adopted to learn temporally consistent appearance. Different from the strategy of implicitly capturing and dealing with motion information, Chan et al. [12] reconsiders the VSR pipeline, and divides it into four essential components, i.e., propagation, alignment, aggregation and up-sampling. Long-term correspondences are exploited based upon bidirectional recurrent propagation from the entire input LR frames for reconstruction. Although it serves as a strong backbone, the misaligned and occluded regions still limit the efficacy of information propagation and aggregation.

Recently, inspired by the relation modeling capabilities of transformer [13] in natural language processing, impressive progresses have been made in image classification [14], object detection [15] and low-level visions [16]. For example, VSR-Transformer [17] tries to use attention mechanisms [18,19] for aligning different frames when dealing with VSR tasks. However, the computational complexity of transformer grows quadratically corresponding to the video frames. Thus, utilizing transformers in videos in proper ways still exists challenges.

To tackle these challenges, we propose a spatial–temporal transformer framework for VSR with multi-scale transformer-based grid propagation and attention-guided deformable motion compensation. The proposed network has two core designs to make it suitable for information propagation and aggregation. The first key element is a stack of spatial–temporal transformer blocks with a multi-scale design. We disentangle the information propagation into two easier sub-tasks with different transformer blocks. In the spatial transformer block, the module exploits the locality information by calculating the attention between all tokens from the same frame. This design helps to focus on enhancing the relevant features within the frame and model the stationary background texture in the same frame. In the temporal transformer block, the extracted tokens are divided into multiple regions, and the module calculates the attention between tokens from different frames' same region and helps to capture past and future information. This module is mainly used to ameliorate information flow and model temporal object movement for improving the robustness of the network against occluded and fine regions. In this way, the two designed transformer blocks attend different missions and are intertwinedly stacked, leading to thorough locality spatial information and temporal data consistency propagation and enhancement. Meanwhile, our transformer operation handles spatial and temporal information separately, making it easier to learn expressive features and get better VSR performance. In addition, a design of multi-scale transformer-based network is able to extract the multi-scale features to tackle various motions.

The second key element is that we explore how to aggregate information from different spatial–temporal locations with the help of feature alignment. Unlike most existing vision recurrent propagation approaches [11] lack feature alignment, the proposed framework consists of a bidirectional propagation with deformable feature alignment. Considering rich dependencies from both the forward and backward frames, the design of bidirectional propagation makes it easier to aggregate information from different locations. Furthermore, deformable alignment [20] helps to align and exploit propagated information in the video sequence. However, the occlusion and fast movement easily influence the accuracy of alignment, and harmful features getting into the propagation process will lead to inferior performance. To address the above limitation, we propose an attention-guided deformable motion compensation and add it to the process of bidirectional propagation. In the proposed module, deformable convolutional offsets are learnt for applying to the unwrapped features, meanwhile, the elementwise correlation between different features at each location is calculated for indicating how informative it is and eliminating inaccurate features after alignment.

By adopting such a pipeline, the proposed approach achieves state-of-the-art performance on VSR with much fewer parameters. In summary, the contributions of our work are threefold:

- We propose a novel multi-scale spatial–temporal transformer framework for VSR to improve the ability of locality spatial and temporal data information propagation. The task of spatial–temporal propagation is divided into two easier sub-tasks to process spatial and temporal information separately.
- We design a bidirectional propagation framework with attention-guided deformable motion compensation. The seamless combination of bidirectional propagation and attention-guided feature alignment benefits VSR performance a lot.
- We conduct experiments on three widely used benchmark datasets and show that the proposed method outperforms previous state-of-the-art VSR methods both qualitatively and quantitatively, especially for recovering the fine details.

Related work

With the help of deep learning, SR which aims to recover HR details from LR images has drawn significant attention. Recently, several attempts under transformer mechanism are proposed to solve SR. In this section, we detail the existing VSR approaches from three aspects: space based, recurrent time based and vision transformer based.

Video super-resolution: Existing VSR approaches can be mainly divided into two frameworks—space based and recurrent time based. Earlier methods [21,22] in the space-based framework predict the optical flow between input LR frames and perform alignment for merging information of multiple continuous frames. Later approaches resort to a more sophisticated approach of implicit alignment. For instance, EDVR [20] adopts multi-scale deformable convolutions for accurate alignment at the feature level. VSR-DUF [23] builds dynamic up-sampling filters to estimate motion implicitly. Wei et al. [24] propose to add non-local operations and dense connections for further alleviating misalignment. Meanwhile, some approaches adopt a recurrent time-based framework, taking the past and future information as temporal supplementary materials. FRVSR [25] sends each LR frame through the network to obtain its super-resolved HR outputs. As frames go through the network, the previous HR outputs are sent back to help to reconstruct the following LR frames. Lai et al. [9] propose to exploit the structures in previously estimated HR frames for guiding the sparse reconstruction of the next LR frame. RSDN [26] adopts a hidden state adaptation module and a recurrent detail-structural block to improve the robustness to error accumulation and appearance change. The aforementioned works have introduced many sophisticated and new components to address the alignment and aggregation problem. Here, we reconsider some remaining issues and find that bidirectional propagation with spatial–temporal transformer and attention-guided deformable motion compensation suffice to outperform many existing methods.

Vision transformers: Recently, transformer improves the performance of natural language processing significantly. Different from the CNNs, the structure of transformer-based framework makes it good at capturing long-term data dependencies. The success of transformer on natural language processing inspires the research on computer vision. Image classification [14], segmentation [27], object detection [15], et al., adopt transformer to explore the global interactions between different regions and learn to attend to important regions.

Inspired by previous works, a standard transformer model is also applied to a backbone model for restoration problem [28], especially for SR. Wang et al. [29] propose a U-shaped network with swin transformer [30] for SR. SwinIR [31] tackles SR task using swin transformer and brings residual operations into swin transformer. In this work, we also compute window and shift window attentions similar to swin transformer. However, building correlations within and between frames under multi-scale design is explored for enhancing stationary and motive textures. In addition, VSR-Transformer [17] and TTVSR [16] try to exploit attention mechanisms for utilizing different frames. Different from their designs of aggregating information, considering the spatial–temporal information in the video is complicated to learn simultaneously, we decouple the task of spatial–temporal propagation into two easier sub-tasks with a multi-scale design, and learn different components by decoupled transformers.

Methods

In this section, we first introduce the proposed VSR network architecture in "Network architecture", and then discuss the proposed attention-guided deformable motion compensation in "Attention-guided deformable motion compensation". Finally, we focus on our design of multi-scale spatial and temporal transformer in "Multi-scale spatial and temporal transformer".

Network architecture

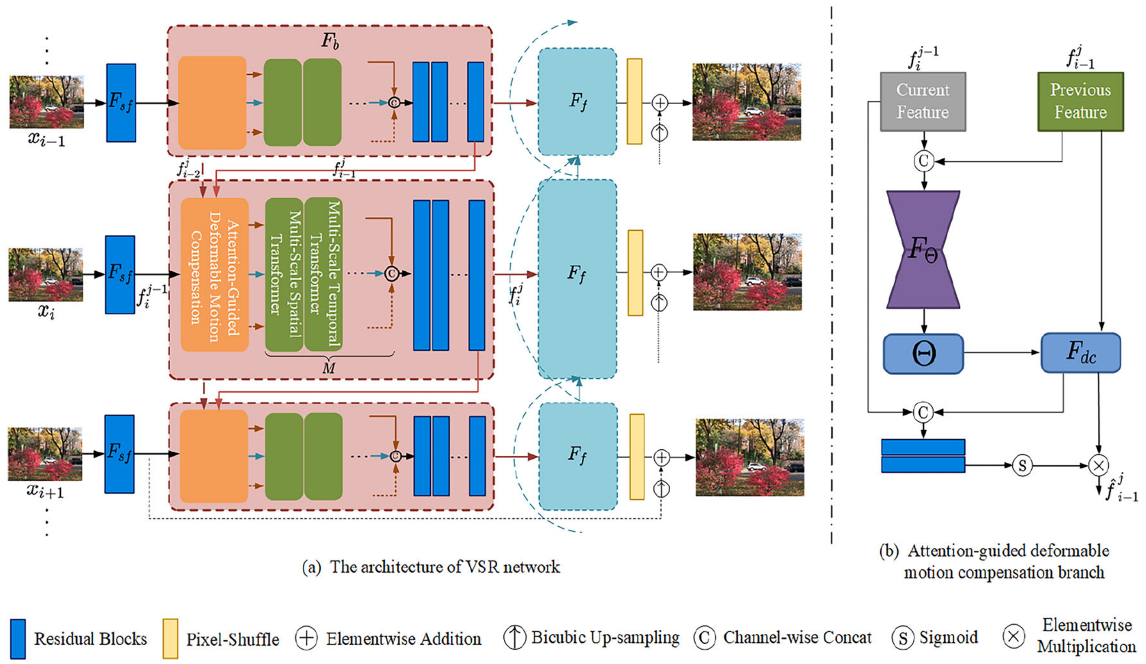
Let $\mathcal{X} = \{x_{i-n}, \cdots, x_i, \cdots, x_{i+n}\}$ denotes an LR video sequence with length $N = 2n + 1$, where x_i means the i-th frame of this input video sequence. We aim to learn a mapping function that takes an LR video $\mathcal{X}$ as input and outputs an HR video $\mathcal{Y}$. Our proposed network is an end-to-end fully trainable framework, which consists of three modules: shallow feature extraction, bidirectional propagation and up-sampling. The overall pipeline is shown in Fig. 1.

Shallow feature extraction: Given an LR input $\mathcal{X} \in \mathbb{R}^{N \times H \times W \times C}$ (H, W and C are the height, width and input channel number of each LR input frame), we use residual blocks $F_{sf}(\cdot)$ first to extract shallow feature $g^0 \in \mathbb{R}^{H \times W \times C_{in}}$ from each frame as

$$g_i^0 = F_{sf}(x_i), \tag{1}$$

where C_{in} denotes the feature channel number, g_i^0 is the extracted shallow features from x_i by multiple residual blocks. The residual blocks are used for early visual processing, and provide a simple way to map the input image space to a higher dimensional feature space.

Bidirectional propagation: After shallow feature extraction, motivated by the effectiveness of bidirectional propagation, we adopt it for exploiting dependencies in the video



(a) The architecture of VSR network

(b) Attention-guided deformable motion compensation branch

Fig. 1 **a** The architecture of our whole network. The network consists of two modifications to improve alignment and propagation. For alignment, we design attention-guided deformable motion compensation to align features and exclude the harmful components that will affect information aggregation. Multi-scale spatial and temporal transformer blocks are adopted to search dependencies between arbitrary pixels across all the spatial–temporal positions. Here, we detail the procedure for backward propagation F_b, forward propagation F_f is defined similarly. In addition, bidirectional (solid lines and dotted lines) connections are adopted to improve the robustness of propagation. Within each forward or backward propagation, multi-scale spatial and temporal transformers are proposed to refine features further. All the feature maps have 64 channels. **b** The details of attention-guided deformable motion compensation branch

sequence. Specifically, our bidirectional propagation module mainly consists of a series of forward and backward operations ($F_f(\cdot)$ and $F_b(\cdot)$), the intermediate features are propagated forward and backward in an alternating manner. Through propagation, the information from different frames can be adopted and aggregated for feature refinement. To further improve the robustness of propagation, dense connections are adopted to assist aggregate information from different temporal positions, improving effectiveness and robustness in fine and occluded regions. To compute the feature f_i^j after backward propagation, the operation $F_b(\cdot)$ will align and concatenate f_i^{j-1}, f_{i-1}^j and f_{i-2}^j, which will be discussed later:

$$f_i^j = F_b(f_i^{j-1}, f_{i-1}^j, f_{i-2}^j), \tag{2}$$

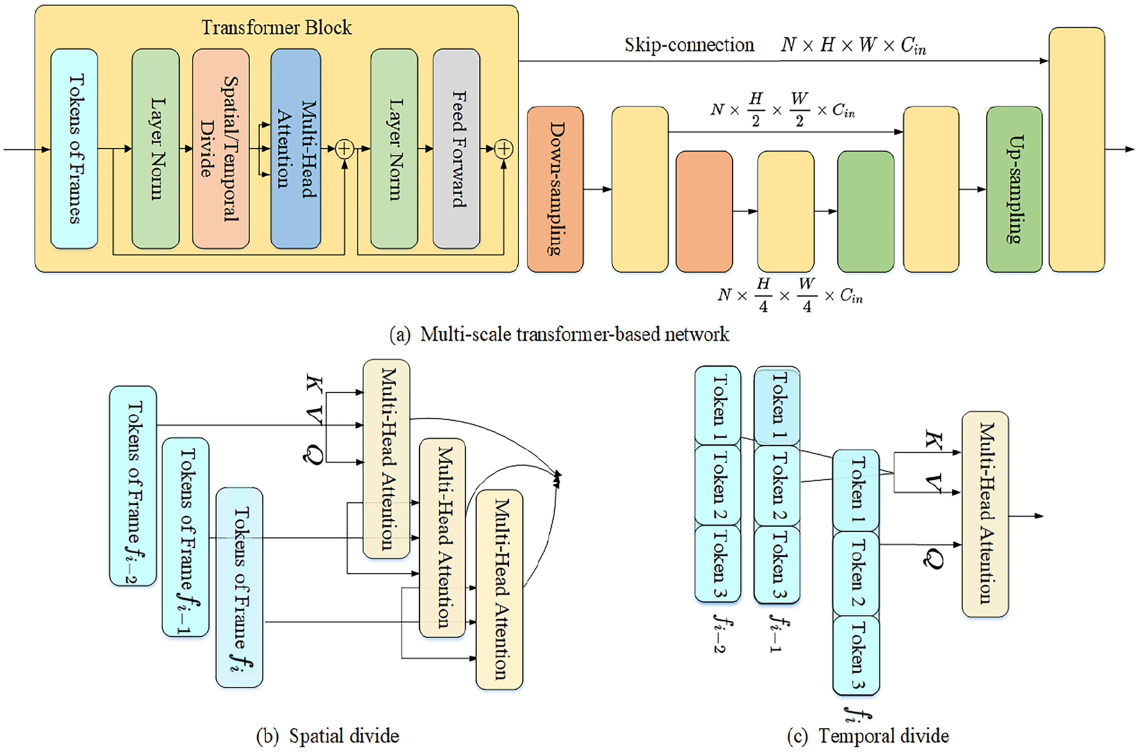
where f_i^j denotes the feature computed at the i-th timestep in the j-th propagation branch.

Up-sampling: In the final step, the aggregated features from the preceding layer are propagated back onto the pixel domain through pixel-shuffling F_u [32], and we generate the output HR image by concatenating the current LR input frame x_i in the channel dimension.

Attention-guided deformable motion compensation

To alleviate the harmful impact of nonalignment on subsequent information aggregation, in each forward or backward propagation, we first adopt a motion compensation module for aligning features. The graphical illustration of backward propagation is shown in Fig. 1a. Deformable convolution [33,34] has demonstrated significant performance on feature alignment due to offset diversity; however, the training process of offset is unstable, which often results in offset overflow, deteriorating subsequent performance. To take advantage of the offset diversity while avoiding instability, the input features are first concatenated and sent to an encoder–decoder module for computing stable offset. To further enhance the robustness of feature alignment, we propose to focus on valuable information and avoid harmful features feeding into the subsequent stages. The architecture of our motion compensation branch is shown in Fig. 1b.

At the i-th timestep, given the target features f_i^{j-1} computed from previous module, the feature f_{i-1}^j computed from the previous timestep, the deformable motion compensation module first concatenates them, then feeds it into encoder–decoder convolutions F_Θ to predict offset Θ for the feature



(a) Multi-scale transformer-based network

(b) Spatial divide

(c) Temporal divide

Fig. 2 **a** The architecture of multi-scale transformer-based network, which extracts multi-scale features to adapt different degrees of motions and details. **b** Illustration of spatial transformer operation. **c** Illustration of temporal transformer operation. According to different transformer ways, divided spatial/temporal multi-head attention and feed-forward module are contained in the transformer block

f_{i-1}^j:

$$\Theta = F_\Theta(f_i^{j-1}, f_{i-1}^j). \tag{3}$$

With Θ and f_{i-1}^j, the aligned feature $\overline{f}_{i-1}^j$ can be computed by the deformable convolution:

$$\overline{f}_{i-1}^j = F_{dc}(f_{i-1}^j, \Theta). \tag{4}$$

Generally, the procedure of feature alignment is completed. More specially, in our module, attention is adopted to further guide and fix aligned features. We concatenate f_i^{j-1} and aligned feature $\overline{f}_{i-1}^j$, and feed it into two 3 × 3 convolution layers. The result after convolutions will be performed sigmoid activation for generating 64-channel attention map $\mathcal{A}$. The computational process can be formulated as:

$$\mathcal{A} = \text{Sigmoid}(F_\mathcal{A}(f_i^{j-1}, \overline{f}_{i-1}^j)), \tag{5}$$

where $F_\mathcal{A}$ represents the attention operation with two 3×3 convolution layers. In this way, $\mathcal{A}$ has the same size as $\overline{f}_{i-1}^j$. The values in $\mathcal{A}$ are in the range [0, 1]. The computed attention map $\mathcal{A}$ is used to attend the features $\overline{f}_{i-1}^j$ via:

$$\hat{f}_{i-1}^j = \mathcal{A} \otimes \overline{f}_{i-1}^j, \tag{6}$$

where $\otimes$ denotes the elementwise multiplication and $\hat{f}_{i-1}^j$ is the final aligned features with attention guidance.

Based on the further attention operation, generated attention maps evaluate the importance of different image regions for the target frame. Dependent information is highlighted, and regions with misalignment and harmful features are excluded, improving robustness and effectiveness in motion and fine regions.

Multi-scale spatial and temporal transformer

To further exploit the locality and spatial–temporal data information from different frames, we design spatial and temporal blocks with a multi-scale design shown in Fig. 2a, which utilizes multi-scale features to handle various degrees of motions and details. To be specific, the multi-scale transformer-based network consists of an encoder and a decoder operation with symmetrical architecture. When encoding in each stage, the transformer block first models the input features, then the features are spatial downsampled two times by the down-sampling operator. The down-sampling operator reshapes the input features with shape $\mathbb{R}^{N \times H \times W \times C_{in}}$ to the shape $\mathbb{R}^{N \times \frac{H}{2} \times \frac{W}{2} \times 4C_{in}}$ and then linearly projects it to $\mathbb{R}^{N \times \frac{H}{2} \times \frac{W}{2} \times C_{in}}$. In each stage of decoder, the up-sampling operator first increases the number of feature channels by linear projection, then reshapes it to spatial upsampled features. The features outputted by the previous decoder stage are then fused with the features from the corresponding encoder stage through skip-connection for further transformer-based feature refinement.

In addition, for better searching and propagating features from different spatial positions and temporal dimensions, we design spatial and temporal transformer blocks used in multi-scale transformer-based network. Spatial transformer is mainly used to calculate dependencies between extracted tokens from the same frame, and temporal transformer is engaged in calculating the dependencies between tokens from different frames' same location. In this way, the two designed transformer blocks attend different missions and are intertwinedly stacked, leading to thorough locality spatial information and temporal data consistency propagation and enhancing. Meanwhile, our transformer operation handles spatial and temporal information separately, making it easier to learn expressive features and get better VSR performance.

Given the feature f_i (the superscript j is omitted for notational simplicity) generated by the previous layer, it is first partitioned into m^2 zones along both the height and width dimension, similar to [35]. $f_i^{k,s}$ denotes each split zone, where $k, s = 1, \ldots, m$. Then, according to different ways of transformer, we group these tokens together. In spatial transformer block, the tokens are grouped along spatial dimension, denoted as $P_i = \left\{ f_i^{1,1}, f_i^{1,2}, \ldots, f_i^{m,m} \right\}$. In this way, spatial transformer block takes each token set P_i as input, and performs multi-head attention across all tokens in it. For each feature $f_i^{k,s}$, a linear projection with transformation matrices W_q, W_k and W_v is performed on it to generate the Query, Key and Value:

$$Q = f_i^{k,s} W_q^Q, K = f_i^{k,s} W_q^K, V = f_i^{k,s} W_q^V, \tag{7}$$

where W_q^Q, W_q^K, W_q^V represent the projection matrices for the q-th head, respectively. Then the multi-head attention is performed. The computational process of the q-th head self-attention in the split zone can be defined as:

$$P_i = \left\{ f_i^{1,1}, f_i^{1,2}, \ldots, f_i^{k,s} \right\}, k, s = 1, \ldots, m, \tag{8}$$

$$Y_i^{k,s} = \text{Attention}(f_i^{k,s} W_q^Q, f_i^{k,s} W_q^K, f_i^{k,s} W_q^V), \tag{9}$$

$$\bar{P}_i = \left\{ Y_i^{1,1}, Y_i^{1,2}, \ldots, Y_i^{m,m} \right\}, \tag{10}$$

$\bar{P}_i$ is the output of the q-th head. Then the outputs for all head are concatenated and then linearly projected to get final result. The attention calculation is formulated based on previous work [30], it is performed as:

$$\text{Attention}\,(Q, K, V) = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d}}\right)V. \tag{11}$$

This helps the model search for similar features in the spatially neighboring pixels and propagate these textures to target regions for compensating incomplete information. Shift window operation in swin transformer [30] is also utilized to enhance the long-range dependency modeling capacities of the transformer.

Another way is to gather these tokens together along temporal dimension $P^{k,s} = f_{i-n}^{k,s}, \ldots, f_{i+n}^{k,s}, k, s = 1, \ldots, m$. All tokens are unused to generate Query, Key and Value, only the tokens from the reference feature are used to generate Query, and other tokens are used to generate Keys and Values. Then, temporal transformer block performs multi-head attention across all tokens in each input $P^{k,s}$. By doing so, long-term dependencies and continuous related movement can be detected and help the motion region for fine recovery. More details of two different transformers are illustrated in Fig. 2b, c.

In both spatial and temporal transformer blocks, after spatial/temporal divided attention module, a feed-forward module follows it. The overall operation of the transformer block is formulated as:

$$\begin{aligned} \hat{P}_i = {} & F_{\text{ffn}}(F_{\text{ln}}(F_{\text{msa}}(F_{\text{sd}}(F_{ln}(P_i))) \\ & + P_i)) + F_{\text{msa}}(F_{\text{sd}}(F_{ln}(P_i))) + P_i, \end{aligned} \tag{12}$$

$$\begin{aligned} \hat{P}^{k,s} = {} & F_{\text{ffn}}(F_{\text{ln}}(F_{\text{msa}}(F_{\text{td}}(F_{ln}(P^{k,s}))) \\ & + P^{k,s})) + F_{\text{msa}}(F_{\text{td}}(F_{\text{ln}}(P^{k,s}))) + P^{k,s}, \end{aligned} \tag{13}$$

where the former equation denotes the spatial transformer block and the latter is the operation of temporal transformer block. F_{ffn} is the feed-forward module, F_{ln} is the layer normalization [36] and F_{msa} is the multi-head attention module. F_{sd} and F_{td} are the spatial and temporal operations, respectively. Note that input varies with different gathering ways generates different searching zones, which helps detect related features and continuous movements to achieve coherent completion.

Experiments

In this section, we first introduce the experimental settings: training datasets, testing datasets, training details, and evaluation metrics in "Experimental settings". Then, ablation studies on different settings of the proposed method are shown in "Ablation studies". Finally, we evaluate the performance of our approach and compare it with the state-of-the-art SR algorithms in "Comparison with the state-of-the-art methods". In all our experiments, we focus on ×4 VSR factor.

Table 1 Quantitative comparisons of different components

	(A)	(B)	(C)	(D)	Ours
Attention-guided deformable motion compensation		√	√	√	√
Bidirectional propagation			√	√	√
Multi-scale operation				√	√
Spatial–temporal transformer					√
PSNR/SSIM	31.18/0.869	31.67/0.881	31.82/0.884	32.09/0.886	32.27/0.887

All scores are the average across the SPMCS [21] testing set. Each component brings significant improvements in different evaluation metrics, indicating their effectiveness

Experimental settings

Training datasets: In the experiment, a public video dataset [37] is adopted as our training set, which contains 64612 7-frame sequences with fixed 448 × 256 resolution and has been widely used in many VSR methods [12,20]. Specially, we downscale the HR frames by bicubic interpolation to generate LR frames and apply several data augmentation techniques to the paired training data for enlarging the training set, such as rotation, flipping and random cropping.

Testing datasets: We test the designed model on three testing sets: Vid4 [38], SPMCS [21] and Vimeo-90k [37]. Vid4 is widely used in literature for comparison, but it includes only four video sequences with limited motions and little inter-frame variations, and has limited ability to assess the relative merits of competing approaches. Although the SPMCS dataset has more variations, it still lacks challenging videos with fast motions. Therefore, we also use a larger Vimeo-90k dataset with rich scenes and motion types as a testing set for evaluating different methods. When comparing, we stratify the sequences of Vimeo-90k into slow, medium and fast queues based on the estimated motion velocities as in [39], and evaluate these queues separately. This makes it easier to reflect the advantages of the proposed approaches.

Training details: We train our model with Adam optimizer [40] by setting $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate of the main network and the attention-guided deformable motion compensation module are initially set to 10^{-4} and 10^{-5}, respectively. The total number of iterations is 600K, and the weights of the motion compensation module are fixed during the first 10 K iterations. We set the batch size as 8 and use Charbonnier penalty function [41] as the final loss since it better improves the performance over the conventional ℓ_2-loss [42]. We perform the forward and backward propagations two times. We repeat a spatial transformer block followed by a temporal transformer block two times in each forward or backward propagation. The number of residual blocks for each propagation branch is set to 5. The channel size is set to 64 and the patch size of input LR frames is 64×64. We implement our models with the PyTorch framework and train them using two NVIDIA 3090 GPUs.

Evaluation metrics: We use PSNR and SSIM as evaluation metrics to compare with other SR networks quantitatively. In the comparison, the estimated first and last two frames are not used for evaluation, and 8 pixels near image boundaries are ignored. All measurements use only the luminance channel (Y).

Ablation studies

To analyze the contributions of the various components (i.e., attention-guided deformable motion compensation, bidirectional propagation, multi-scale operation, spatial and temporal transformer), we start with a baseline and gradually insert different components. Table 1 shows that each component brings considerable improvement, ranging from 0.18 to 1.09 dB in PSNR.

Attention-guided deformable motion compensation

We further provide some visual comparisons of intermediate results to verify the contributions of the proposed attention-guided deformable motion compensation module. As shown in Fig. 3, we compare the computed deformable convolutional offsets of different images. The deformable motion compensation module produces offsets that highly reflect the motions between frames. The refined motion estimation allows deformable convolution to receive information from multiple positions, providing more flexibility.

In addition to motion compensation, in our module, attention is adopted to further guide and fix aligned features. Figure 3d and h are the generated features before and after attention module, respectively. When receiving the adjacent propagated features, the aligned features still contain some misaligned edges and textures, which are still harmful to information aggregation. In contrast, by further highlighting related information and excluding harmful features through attention, the feature shows sharper and preserves more details. Figure 4 shows the generated attention maps of neighboring images, the misaligned regions can be excluded efficiently (i.e., the edges of moving car, pedestrian and the

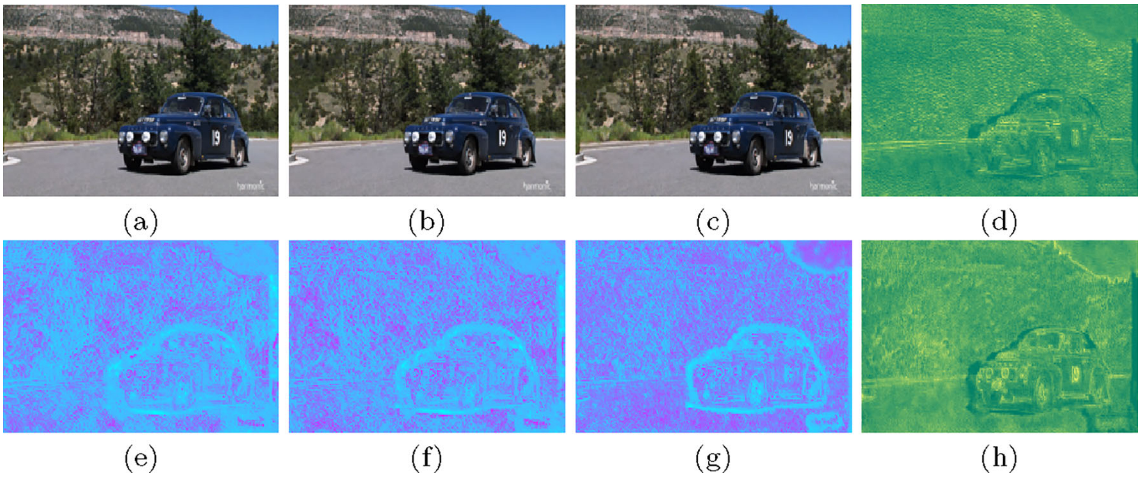


Fig. 3 Analysis of attention-guided deformable motion compensation. **a**–**c** The sequences adjacent to the reference image. **e**–**g** The computed deformable convolutional offsets of continuous images. **d** Generated feature before attention module. **h** Generated feature after attention module

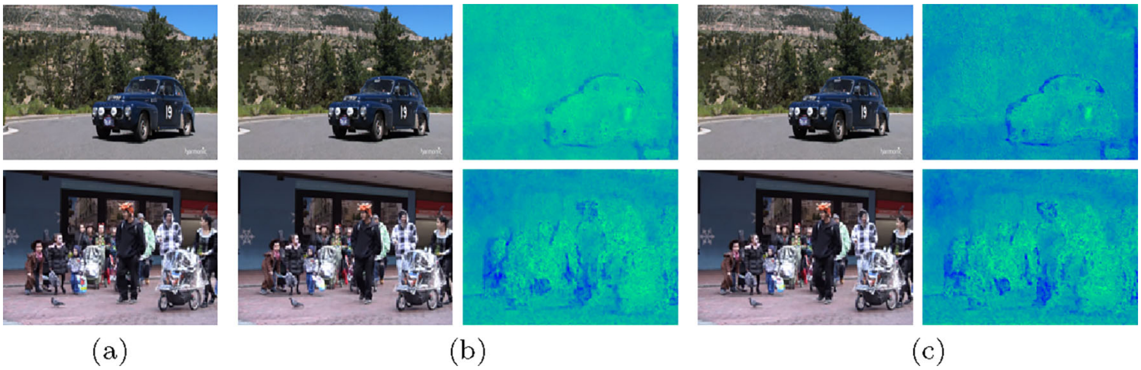


Fig. 4 Comparisons of generated attention map. **a** Reference image. **b**, **c** Neighboring image and corresponding attention map

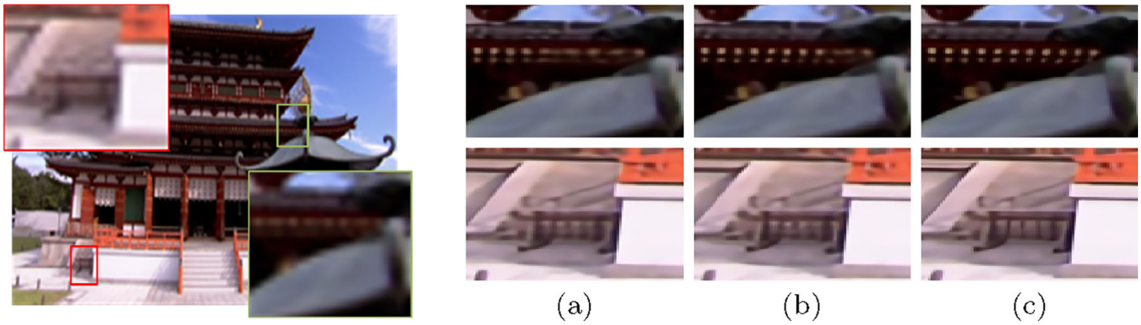


Fig. 5 Comparisons of multi-scale spatial and temporal components. **a** The results outputted from the proposed network without temporal transformer components. **b** The results outputted from the proposed network without spatial transformer components. **c** The results outputted from the proposed network with all transformer components. Results based on only spatial transformer component perform better on reconstructing stationary texture and the results based on only temporal transformer component perform better on recovering the region with motion and occluded object. The emsembling results show better robustness and quality

scene change) and useful features are highlighted. This further verifies the effectiveness of our attention module.

Multi-scale spatial and temporal transformer

To understand the contributions of the proposed multi-scale spatial and temporal transformer components, we design an ablative comparison by removing the intermediate temporal

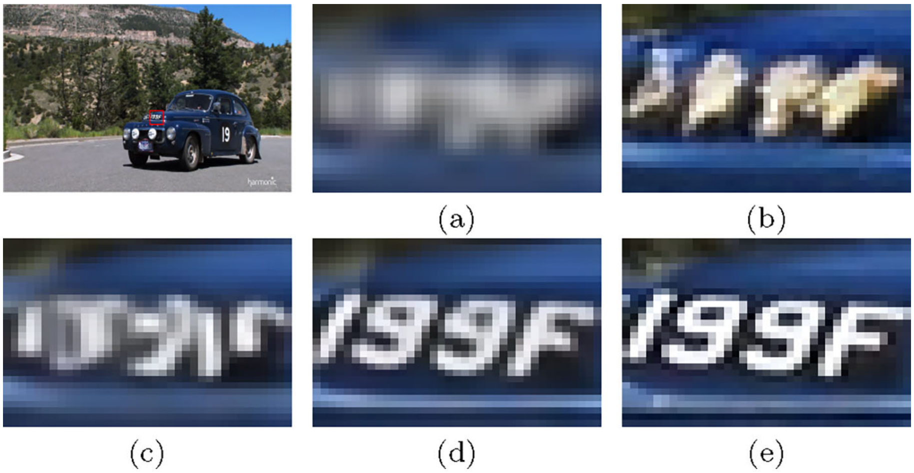


Fig. 6 ×4 VSR using multiple frames. **a** Bicubic. **b** SwinIR [31]. **c** The reconstructed result using copied frames. **d** The reconstructed result using consecutive frames. **e** Ground truth. The comparison demonstrates the recovered textures truly come from the original input LR frames

and spatial transformer branches separately. In Fig. 5a, the recovered video from the network with only a multi-scale spatial transformer component performs better stationary texture by searching and detecting similar details from neighboring regions in the same frame. The network with only a multi-scale temporal transformer component performs better on the regions with occlusion. The proposed network combined with all spatial and temporal transformer branches shows better performance and effectiveness.

Detail fusion vs. synthesis

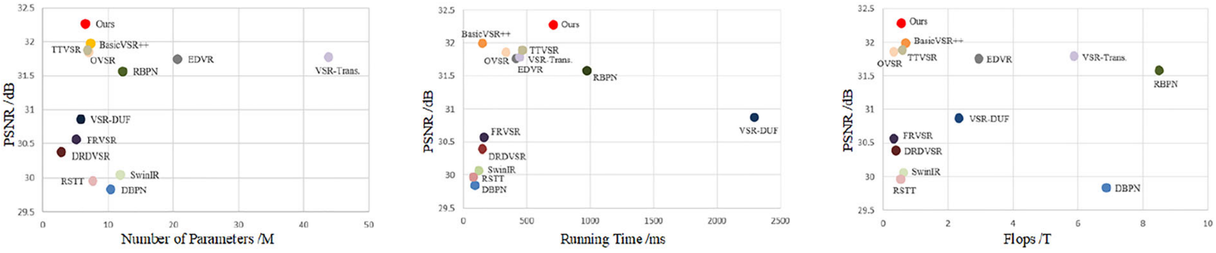
We also analyze whether the recovered details truly come from the input sequences or exist in external data. Here, we replace the inputs with the copied frames and design a more illustrative experiment. As shown in Fig. 6d, when using sequence frames as inputs, it shows that the details are recovered nicely. However, if we use copied frames to evaluate the same model, the result is almost the same as that recovered using just one frame with SwinIR [31] as one of SISR methods (see Fig. 6b, c). All of these comparisons demonstrate that the recovered information shown in Fig. 6d truly generate from different input frames, and they are not synthesized from external examples. Because if it is generated by synthesis, the recovered details will also appear when we use the copied frames. All of these show that the proposed model can exploit internal details and merge them to recover more textures.

Comparison with the state-of-the-art methods

We conduct comprehensive experiments by comparing with the current state-of-the-art SISR and VSR algorithms: SwinIR [31], DBPN [43], DRDVSR [21], FRVSR [25], VSR-DUF [23], RBPN [39], EDVR [20], VSR-Trans. [17], OVSR [44], RSTT [45], TTVSR [16] and BasicVSR++ [46]. Notice that, under our training dataset, only VSR-DUF, RBPN, EDVR, OVSR, RSTT and TTVSR methods provide their pretrained model. In order to make a fair comparison with other approaches, we have retrained other methods, e.g., SwinIR, DBPN, DRDVSR, FRVSR and BasicVSR++ based on their provided codes, using the same training datasets, with same down-sampling scheme.

We first conduct experiments on parameter numbers, testing time costs and Flops by comparing ours with different models. The results about parameter numbers are shown in Fig. 7a. EDVR and VSR-Trans. expect to use more layers to learn the mappings between input frames and the final HR image. Our model conducts a recurrent framework for learning dependencies, and our method achieves better performance. The comparisons of running time and PSNR values are shown in Fig. 7b. VSR-DUF has a significantly larger running time because it needs to perform many 3D convolution operations and large matrix multiplication. It takes about 680 ms for our method to generate one 528 × 926 frame under ×4 SR, while VSR-DUF costs about 2270 ms. At the same time, our approach can achieve the best quantitative performance. Compared with BasicVSR++, although we cost more time due to the introduction of the transformer mechanism, which replaces convolution operation of higher parameter with a large scale of matrix multiplication [31], our approach achieves higher performance while keeping a comparable number of parameters. As shown in Fig. 7c, the Flops are computed with the input of LR frames. It should be emphasized that our approach is lighter. Such superior performance mainly benefits from the use of decoupled transformer design, which significantly reduces the parameters and computational costs.

Second, we perform experiments on different testing datasets with other super-resolution approaches. As can be seen in Table 2, the quantitative comparisons on Vid4 [38] indicate that our approach can obtain the best quantitative results of PSNR/SSIM when comparing with the previous state-of-the-art VSR and SISR models. In addition, based on



(a) (b) (c)

Fig. 7 Parameters, speed, Flops and performance comparison. **a** Parameter numbers and performances of various methods. **b** Testing time costs. (generating one 528 × 926 frame when the upscaling factor is 4) and performances. **c** Flops and performances of various methods

Table 2 PSNR/SSIM of SR methods on Vid4 [38] with upscale 4

Method	Calendar	City	Foliage	Walk	Average
Flow magnitude	1.14	1.63	1.48	1.44	1.42
SwinIR [31]	21.68/0.710	25.82/0.696	24.64/0.682	28.62/0.855	25.19/0.735
DBPN [43]	21.59/0.703	25.41/0.684	24.07/0.662	28.01/0.839	24.77/0.722
DRDVSR [21]	22.26/0.751	27.07/0.758	25.59/0.724	29.01/0.878	25.98/0.777
FRVSR [25]	23.78/0.788	27.32/0.796	25.78/0.748	29.67/0.898	26.63/0.807
VSR-DUF [23]	23.88/0.793	27.76/0.805	26.23/0.754	30.60/0.908	27.11/0.815
RBPN [39]	23.99/0.807	27.64/0.802	26.27/0.757	30.65/0.911	27.16/0.819
EDVR [20]	24.10/0.815	27.96/0.810	26.33/0.762	31.02/0.913	27.35/0.825
VSR-Trans. [17]	24.04/0.812	27.94/0.810	26.33/0.763	31.10/0.916	27.36/0.825
OVSR [44]	24.12/0.815	28.04/0.814	26.45/0.764	31.04/0.915	27.41/0.827
RSTT [45]	23.81/0.781	27.38/0.790	25.82/0.742	29.70/0.891	26.67/0.801
TTVSR [16]	24.10/0.815	27.98/0.814	26.45/0.764	31.07/0.915	27.40/0.827
BasicVSR++ [46]	*24.17/0.816*	**28.08**/*0.815*	*26.40/0.764*	*31.20/0.917*	*27.46/0.828*
Ours	**24.25/0.818**	*28.06*/**0.815**	**26.51/0.766**	**31.25/0.917**	**27.52/0.829**

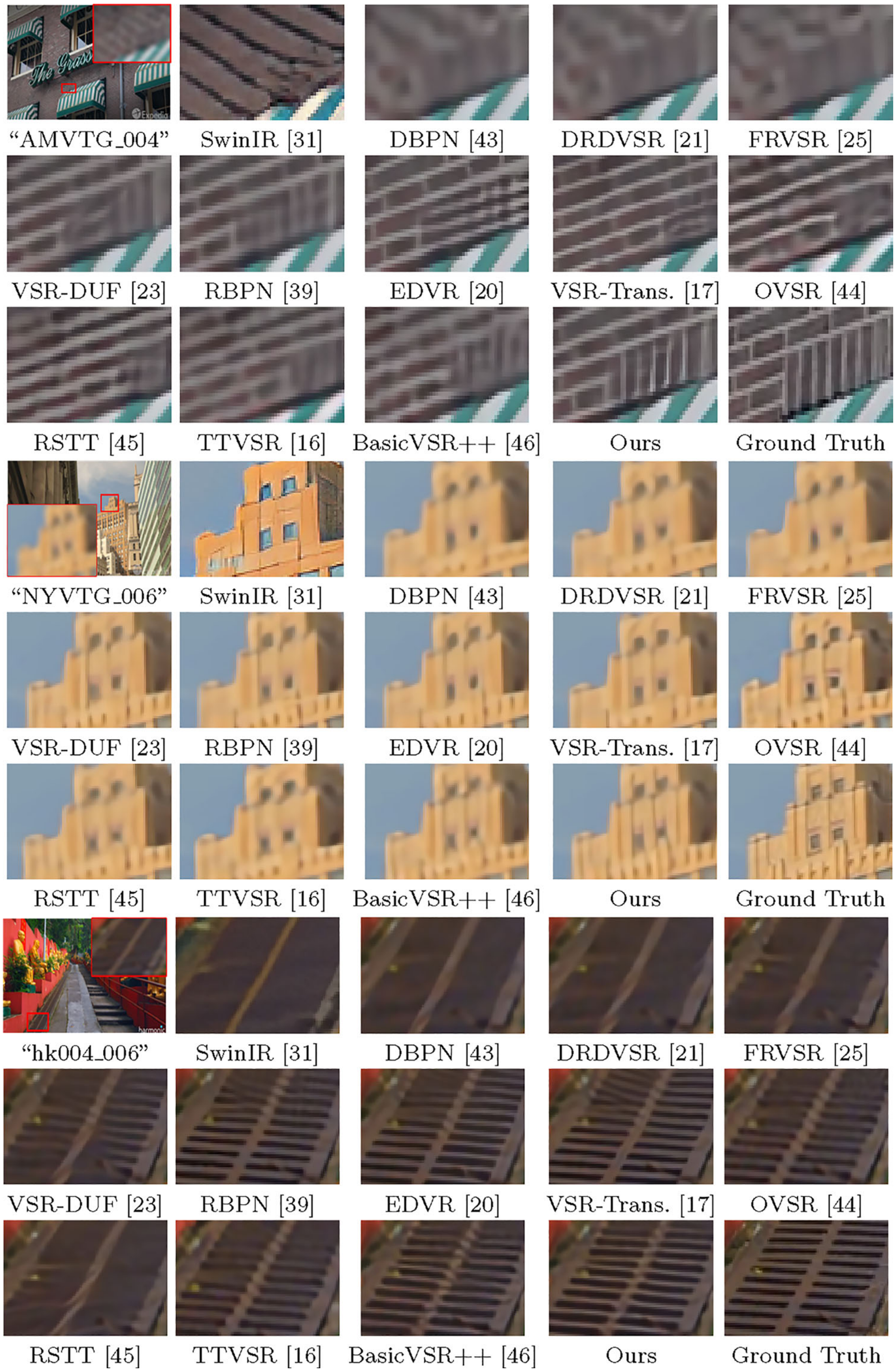
Bold indicates the best and Italic indicates the second best performance

Table 3 PSNR/SSIM of SR methods on SPMCS-11 [21] with upscale 4

Method	car05	hdclub	hitachi	hk004	HKVTG	jvc	NYVTG	PRVTG	RMVTG	veni3	veni5	Average
Flow magnitude	6.21	0.70	3.01	0.49	0.11	1.24	0.10	0.12	0.18	0.36	0.36	1.17
SwinIR [31]	29.88	20.51	23.55	31.68	28.69	28.11	30.25	26.52	25.99	35.07	31.17	28.31/0.827
DBPN [43]	29.58	20.22	23.47	31.59	28.67	27.89	30.13	26.36	25.77	34.54	30.89	28.10/0.820
DRDVSR [21]	30.13	20.98	23.78	32.13	28.80	28.60	31.05	26.86	26.14	35.18	31.55	28.65/0.834
FRVSR [25]	30.62	21.51	24.76	32.12	28.95	28.98	31.63	27.24	27.00	33.60	30.96	28.85/0.851
VSR-DUF [23]	31.07	21.77	25.83	32.99	29.21	29.62	32.57	27.31	27.63	35.24	31.95	29.56/0.867
RBPN [39]	31.92	21.88	26.40	33.31	29.43	30.26	33.25	27.60	27.69	36.53	32.82	30.10/0.874
EDVR [20]	32.22	22.16	26.86	33.42	29.47	30.44	*33.70*	27.70	27.72	37.07	32.96	30.28/0.876
VSR-Trans. [17]	32.35	22.29	26.98	33.57	29.54	30.59	33.62	27.84	27.81	37.12	33.05	30.43/0.876
OVSR [44]	32.41	22.35	27.07	33.61	29.68	30.69	33.70	27.91	27.94	37.13	33.10	30.50/0.875
RSTT [45]	30.54	21.17	24.11	32.37	29.01	29.05	31.59	27.36	27.28	34.59	31.44	28.95/0.856
TTVSR [16]	32.44	22.39	27.01	33.54	29.78	30.72	33.62	27.99	28.01	37.14	33.16	30.52/*0.879*
BasicVSR++ [46]	*32.51*	**22.50**	*27.19*	*33.74*	**29.82**	*30.89*	33.68	**28.16**	*28.05*	*37.18*	*33.24*	*30.63*/0.877
Ours	**32.57**	*22.44*	**27.49**	**33.79**	*29.79*	**31.05**	**34.22**	*28.06*	**28.07**	**37.32**	**33.58**	**30.76/0.881**

Bold indicates the best and Italic indicates the second best performance. Our approach achieves the best quantitative performance

Fig. 8 Visual comparisons for ×4 SR on the testing data from SPMCS [21]. Local areas full of details and textures are zoom in. The proposed network produces high-quality super-resolved images, especially edges and textures



the statistics of flow magnitude, we find that diverse motion types are not contained in Vid4 dataset. And [39] also mentions that because of JPEG compression, the ground truth in Vid4 contains artifacts and aliasing, this apparently leads in some cases penalizing sharper SR predictions. All of these may lead to an insignificant improvement when comparing with the state-of-the-art methods. Thus, we also conduct more comparisons on various testing sets.

Table 3 shows more quantitative comparisons of different approaches on SPMCS-11 [21] dataset. VSR-Trans. and TTVSR introduce transformer mechanisms in sliding window, which obtains significant improvement over most approaches. However, effective utilization of spatial–

Table 4 Quantitative comparison on Vimeo-90k [37] testing set with upscale 4

Method	Slow	Medium	Fast
Flow magnitude	0.6	2.5	8.3
SwinIR [31]	33.12/0.903	35.55/0.926	37.79/0.942
DBPN [43]	32.98/0.901	35.39/0.925	37.46/0.944
DRDVSR [21]	33.24/0.906	35.78/0.931	37.89/0.944
FRVSR [25]	33.20/0.907	36.24/0.940	38.06/0.947
VSR-DUF [23]	33.86/0.911	36.72/0.946	38.37/0.950
RBPN [39]	34.18/0.920	37.28/0.947	40.03/0.960
EDVR [20]	34.45/0.925	37.75/0.951	40.69/0.963
VSR-Trans. [17]	34.52/0.926	37.79/0.951	40.78/0.962
OVSR [44]	34.50/0.925	37.81/0.951	40.81/0.963
RSTT [45]	33.67/0.910	36.19/0.941	37.53/0.944
TTVSR [16]	*34.62*/0.926	37.83/*0.952*	40.84/*0.963*
BasicVSR++ [46]	34.58/*0.926*	*37.88*/0.951	*40.85*/0.962
Ours	**34.76/0.927**	**38.09/0.953**	**41.05/0.964**

Bold and Italic indicates the best and the second best performance, respectively. Clip "Slow" includes 1616 sequences. Clip "Medium" includes 4983 sequences. Clip "Fast" includes 1225 sequences

temporal dependencies is ignored. BasicVSR++ explores hidden states to model the information of sequence. Nonetheless, the capability of long-range learning is limited by its vanishing gradient issue. Different from them, our method utilizes the spatial and temporal transformer strategy to improve the ability of locality spatial and temporal data information bidirectional propagation. Thus, we outperform existing state-of-the-arts. The qualitative results are consistent with the quantitative evaluations and are shown in Fig. 8. The proposed attention-guided motion compensation helps the network get rid of unwrapped features and attend informative propagation on different frames. All of these improve feature expressiveness and the effectiveness of reconstruction in fine and occluded regions. Such obvious comparisons demonstrate the strong capability and efficiency of our model in VSR.

In addition, we further compare the proposed model with others on Vimeo-90k [37], the quantitative results are shown in Table 4. Our method still shows an improvement of 0.18 dB, 0.21 dB and 0.20 dB over the second best method on slow, medium, and fast clips, respectively. Such compar-

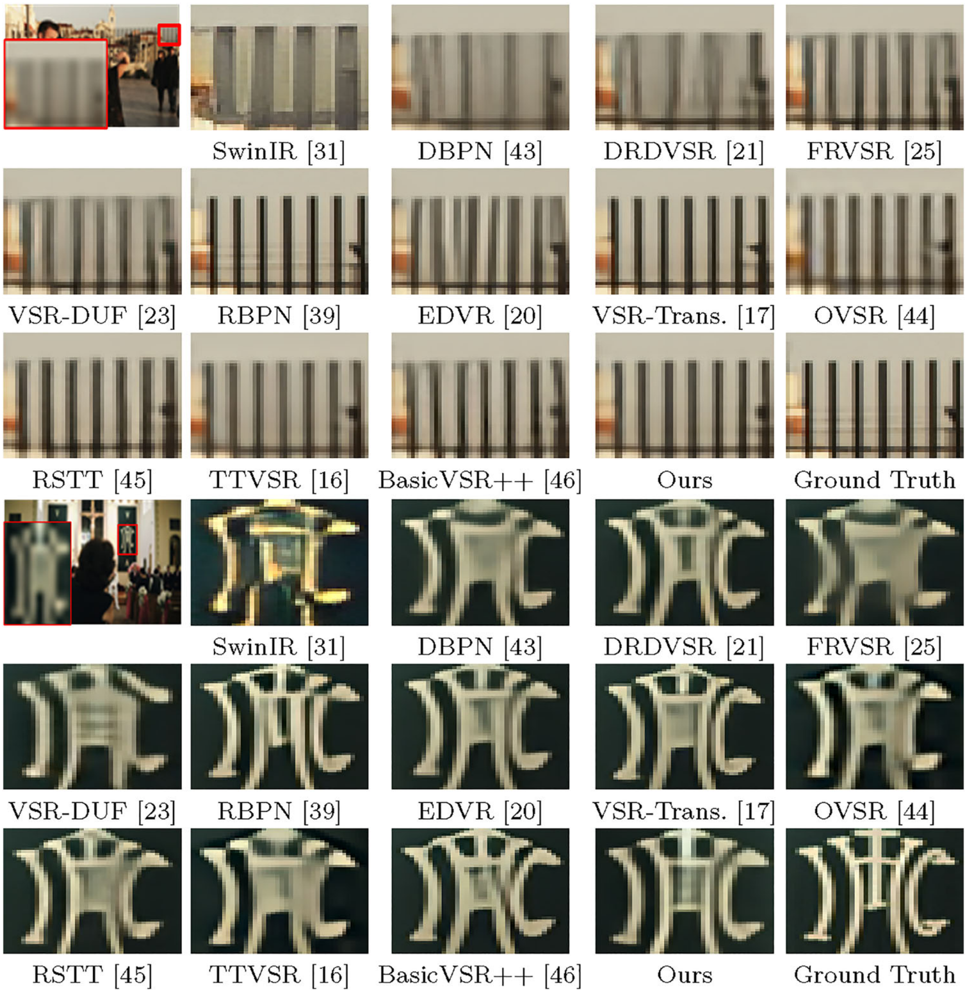


Fig. 9 Visual comparisons for ×4 SR on the testing data from Vimeo-90k [37]. The proposed model is able to recover the correct structures and fine details

isons demonstrate our model can deal with different cases with various motions and further show the effectiveness of the proposed network. Some visual comparisons are shown in Fig. 9, we can find that VSR methods outperform the SISR method overall, because only limited information can be used in SISR. But some error structures and details still appear in the outputs of most VSR methods. In contrast, our approach employs an attention module to help eliminate harmful features in motion compensation, and the proposed multi-scale spatial–temporal transformer framework is able to aggregate locality spatial and temporal data information, all of these improve robustness and effectiveness in fine details and dynamic regions.

Conclusion

In this paper, we study VSR by leveraging spatial–temporal dependencies in LR frames. In particular, we propose a novel end-to-end bidirectional recurrent architecture, which is one of the works to introduce transformer architectures in VSR tasks. Specifically, we formulate video frames into aligned features, and calculate attention along features to prevent harmful information. To implement spatial–temporal fusion in an effective way, we decouple it into two easier sub-tasks and enable transformers to model local and long-range information in videos. Experimental results on various videos with different motion types, scenes and input sizes indicate that our approach can achieve acceptable performance when processing various VSR tasks.

Although our approach achieves comparable performance when comparing with current methods in most cases, it still performs a little worse for the dense textures. This may be due to the reason that our training set does not include enough examples of such dense textures and the motion handling requires to be further improved. Another limitation is our computing time cost, but we will further accelerate our model by trying more strategies, for example, network compression and feature distillation. Moreover, we will further work on exploring how to extract useful features, improve training algorithms and reduce computational redundancy in VSR.

Acknowledgements This work was supported by grants Natural Science Foundation of China (Nos. 62201458, 61901384, 61231016), Natural Science Foundation of Shaanxi Province (No. 2022JQ-577), Special Construction Fund for Key Disciplines of Shaanxi Provincial Higher Education and National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology. And Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Che Aminudin MF, Suandi SA (2021) Video surveillance image enhancement via a convolutional neural network and stacked denoising autoencoder. Neural Comput Appl 34:1–17
2. Kim SY, Oh J, Kim M (2019) Deep SR-ITM: joint learning of super-resolution and inverse tone-mapping for 4k UHD HDR applications. In: IEEE International Conference on Computer Vision, pp 3116–3125
3. Sun W, Sun J, Zhu Y, Hu Y, Ding C, Li H, Zhang Y (2019) Complementary coded aperture set for compressive high-resolution imaging. Neurocomputing 358:177–187
4. Sun W, Gong D, Shi Q, van den Hengel A, Zhang Y (2021) Learning to zoom-in via learning to zoom-out: real-world super-resolution by generating and adapting degradation. IEEE Trans Image Process 30:2947–2962
5. Goyal B, Lepcha DC, Dogra A, Wang S-H (2022) A weighted least squares optimization strategy for medical image super resolution via multiscale convolutional neural networks for healthcare applications. Complex Intell Syst 8(4):3089–3104
6. Park SC, Park MK, Kang MG (2003) Super-resolution image reconstruction: a technical overview. IEEE Signal Process Mag 20(3):21–36
7. Yi P, Wang Z, Jiang K, Jiang J, Ma J (2019) Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In: IEEE International Conference on Computer Vision, pp 3106–3115
8. Sun W, Zhang Y (2020) Attention-guided dual spatial-temporal non-local network for video super-resolution. Neurocomputing 406:24–33
9. Lai Q, Nie Y, Sun H, Xu Q, Zhang Z, Xiao M (2020) Video super-resolution via pre-frame constrained and deep-feature enhanced sparse reconstruction. Pattern Recogn 100:107–139
10. Sun W, Gong D, Shi JQ, van den Hengel A, Zhang Y (2022) Video super-resolution via mixed spatial-temporal convolution and selective fusion. Pattern Recogn 126:108577
11. Fuoli D, Gu S, Timofte R (2019) Efficient video super-resolution through recurrent latent space propagation. In: International Conference on Computer Vision Workshops, pp 3476–3485
12. Chan KCK, Wang X, Yu K, Dong C, Loy CC (2021) Basicvsr: the search for essential components in video super-resolution and beyond. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 4947–4956
13. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, vol 30
14. Vaswani A, Ramachandran P, Srinivas A, Parmar N, Hechtman B, Shlens J (2021) Scaling local self-attention for parameter efficient visual backbones. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 12894–12904

15. Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, Pietikäinen M (2020) Deep learning for generic object detection: a survey. Int J Comput Vis 128(2):261–318
16. Liu C, Yang H, Fu J, Qian X (2022) Learning trajectory-aware transformer for video super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 5687–5696
17. Cao J, Li Y, Zhang K, Van Gool L (2021) Video super-resolution transformer, arXiv preprint arXiv:2106.06847
18. Xing H, Xiao Z, Zhan D, Luo S, Dai P, Li K (2022) Selfmatch: robust semisupervised time-series classification with self-distillation. Int J Intell Syst 37:8583–8610
19. Wu S, Song X, Feng Z (2021) MECT: multi-metadata embedding based cross-transformer for Chinese named entity recognition. Association for Computational Linguistics, pp 1529–1539
20. Wang X, Chan KCK, Yu K, Dong C, Loy CC (2019) EDVR: video restoration with enhanced deformable convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, p 0–8
21. Tao X, Gao H, Liao R, Wang J, Jia J (2017) Detail-revealing deep video super-resolution. In: IEEE International Conference on Computer Vision, pp 4482–4490
22. Xue T, Chen B, Wu J, Wei D, Freeman WT (2019) Video enhancement with task-oriented flow. Int J Comput Vis 127(8):1106–1125
23. Jo Y, Oh SW, Kang J, Kim SJ (2018) Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 3224–3232
24. Sun W, Sun J, Zhu Y, Zhang Y (2020) Video super-resolution via dense non-local spatial-temporal convolutional network. Neurocomputing 403:1–12
25. Sajjadi MSM, Vemulapalli R, Brown M (2018) Frame-recurrent video super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 6626–6634
26. Isobe T, Li S, Jia X, Yuan S, Slabaugh G, Xu C, Li Y-L, Wang S, Tian Q (2020) Video super-resolution with temporal group attention. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 8008–8017
27. Zheng S, Lu J, Zhao H, Zhu X, Luo Z, Wang Y, Fu Y, Feng J, Xiang T, Torr PH et al (2021) Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 6881–6890
28. Chen H, Wang Y, Guo T, Xu C, Deng Y, Liu Z, Ma S, Xu C, Xu C, Gao W (2021) Pre-trained image processing transformer. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 12299–12310
29. Wang Z, Cun X, Bao J, Liu J (2021) Uformer: a general u-shaped transformer for image restoration. arXiv preprint arXiv:2106.03106
30. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030
31. Liang J, Cao J, Sun G, Zhang K, Van Gool L, Timofte R (2021) Swinir: image restoration using swin transformer. In: IEEE International Conference on Computer Vision, pp 1833–1844
32. Shi W, Caballero J, Huszar F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 1874–1883
33. Tian Y, Zhang Y, Fu Y, Xu C (2020) TDAN: temporally-deformable alignment network for video super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 3357–3366
34. Chan KC, Wang X, Yu K, Dong C, Loy CC (2020) Understanding deformable alignment in video super-resolution. arXiv preprint arXiv:2009.07265
35. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S et al (2010) An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
36. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv preprint arXiv:1607.06450
37. Xue T, Chen B, Wu J, Wei D, Freeman WT (2019) Video enhancement with task-oriented flow. Int J Comput Vis 127(8):1106–1125
38. Liu C, Sun D (2014) On Bayesian adaptive video super resolution. IEEE Trans Pattern Anal Mach Intell 36(2):346–360
39. Haris M, Shakhnarovich G, Ukita N (2019) Recurrent back-projection network for video super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 3897–3906
40. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: International Conference on Learning. Representations
41. Charbonnier P, Blanc-Feraud L, Aubert G, Barlaud M (1994) Two deterministic half-quadratic regularization algorithms for computed imaging. In: International Conference on Image Processing, vol 2, pp 168–172
42. Lai W, Huang J, Ahuja N, Yang M (2017) Deep Laplacian pyramid networks for fast and accurate super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 5835–5843
43. Haris M, Shakhnarovich G, Ukita N (2018) Deep back-projection networks for super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 1664–1673
44. Yi P, Wang Z, Jiang K, Jiang J, Lu T, Tian X, Ma J (2021) Omniscient video super-resolution. In: IEEE International Conference on Computer Vision, pp 4429–4438
45. Geng Z, Liang L, Ding T, Zharkov I (2022) Rstt: real-time spatial temporal transformer for space-time video super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 17441–17451
46. Chan KC, Zhou S, Xu X, Loy CC (2022) Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 5972–5981